Proceedings of ICSP2018

Image Steganalysis via Multi-Column Convolutional Neural Network

Qi Ke, Liu DongMing,Zhang Daxing
School of Computer Science, Guangzhou University,Guangzhou, China
Email: qikersa@163.com, Liudm@163.com, dxzhang@hdu.edu.cn

***Abstract*—Deep learning that jointly studies and extracts features is very promising for steganalysis. In this article, we design a simple but effective Multi-column Convolutional Neural Network (MCNN) based on steganalysis architecture for images. The proposed MCNN architecture allows the input image to be of arbitrary size or resolution. In particular, by utilizing filters with receptive fields of different sizes, the features learned by each column CNN are adaptive to variations in payloads. Comprehensive experiments on standard dataset show that MCNN model can detect the state of arts steganographic algorithms with a high accuracy. It also outperforms several recently proposed CNN-based steganalyzers in conditions of the same embedding key stego and cover-source mismatch scenarios.**

Keywords—image steganalysis; MCNN; the same key stego; cover-source mismatch

I. INTRODUCTION

Steganalysis analyses the existence of secret messages in digital media. Practically, the most current steganalysis frameworks involve a large number of techniques for feature extraction and classification. The objective of the feature extraction is to capture as much stego information as possible. Image features such as Rich Models (RM) for spatial domain [1, 2] and transform domain [3, 4] are proposed for stegnalysis. The machine learning based steganalysis classfies stego images from cover ones according to the selected features. Many classification tools have also been published such as Ensemble Classifier [5], SVM [6] and Multilayer Perceptron [7].

As a state-of-the-art classification method, the main advantage of deep learning is the automatic features extraction of the input data to improve the learning of the targeted task [1]. Deep learning architectures, including DBN (Deep Belief Network), CNN (Convolutional Neural Networks) and their variants have been applied with great success in various areas such as character recognition, image classification.

The first CNN-based steganalysis experiment [1], which applied the stacked convolutional auto-encoders, has not reached the accuracy level of SRM steganalysis. Further works [9, 10] improve the steganalysis performance in the context of the "the same embedding key" scenario, which resulted in weakening of security for embedding several messages with the same key. However, some limitations have still to be overcome: varisized image steganalysis and those ones with different payload values (especially smaller ones).

In this paper, a Multi-Column Convolutional Neural Network (MCNN) based steganalyzer is designed to improve the steganlysis performance, which aims at overcoming the limitations noted previously. In summary, the main contributions of this paper are listed as follows:

Firstly, the key idea of the proposed MCNN framework is the use of three columns filters with receptive fields of different sizes (large, medium, small) so that the overall MCNN framework is robust to different stego fields.

Secondly, a convolution layer with the filter size of 1×1 is used before the fully connected layer. Therefore, the input image can be of arbitrary size so that the proposed method is adaptive to large variation in stego image size.

II. RELATED WORKS

The deep learning-based steganalyzer has become a breakthrough technology which outperforms conventional steganalysis methods since the first stacked convolutional auto-encoders structure steganalyer [8].

Qian et al. [9] proposed a CNN with 5 convolutional layers followed by three fully connected layers: two hidden layers of 128 ReLU neurons each and one output layer of 2 softmax neurons. Rather than directly processing the input image, this CNN processes a 252×252 high-pass filtered image issued by a 5×5 kernel. The experiments showed that the proposed CNN only slightly outperformed the state-of-art SRM steganalyzer.

Pibre et al. [10] improved the steganalysis performance in the scenario of reusing the same embedding key. They designed a CNN with 2 layers: 64 kernels and 16 kernels for each layer. The experiments showed that the proposed method had 16% improvement compared with the SRM steganalyzers in case of S-UNIWARD at 0.4bpp embedding rate, but gained bad results in the scenario of different embedding key.

Couchot et al. [11] also designed a CNN-based steganalyzer in the scenario of unique key stego. The architecture took 512×512 image as input image, and the input image was filtered by a single kernel of size 3×3, followed by a layer of 64 filters with more larger kernel. The experiments showed that the proposal defeated many state-of-art steganalysis in case of the "same embedding key".

III. MULTI-COLUMN CNN FOR IMAGE STEGANALYSIS

A. CNN structure

A convolutional neural network (CNN) is designed to automatically extract feature maps with different size kernels, which usually consists of several kinds of layers, namely convolutional layer, fully connected layer.

The convolutional layer is the most important layer in CNN, which usually builds feature maps with three steps: 1) convolutional step, which performs a filtering process using K kernels resulting in K new feature maps, 2) activation step, which adds some nonlinearity to feature maps, 3) pooling step, which is applied to feature reduction by computing the mean value or the max value over regions.

As a classification network, the last convolution layer is usually connected to a fully connected layer. The softmax function works on the fully connected layer in order to

normalize the outputs delivered by the network between [0, 1]. The predicted probability of belonging to which class is given by the softmax function. The classification decision is finally obtained by returning the class with the highest probability.

B. Architecture of the proposed MCNN

Steganography embeds the secret information by selecting random pixels. Consequently, it is better to use large convolution filters to capture the slight modifications performed by stego. Different filter sizes had been proposed such as 3×3, 6×6, 12×12 or 15×15. Overall, the larger the filters are, the more complex stego information the feature map can contain.

Due to the random embedding position and different embedding rate, the patch of images usually contain stego pixels of very different sizes, Hence, filters with receptive fields of the same size are unlikely to capture characteristics of stego pixels at different scales[12]. Therefore, it is more natural to use filters with different sizes of local receptive field to capture the features from the raw patches to the stego patches[12].

Motivated by the success of Multi-column CNN [12], and after some preliminary experiments, we proposed to use the Multi-column CNN (MCNN) to capture stego features. In our MCNN model, the different sizes of filters of each column are used to model the stego patches corresponding to stego pixels of different sizes. For instance, filters with larger receptive fields are more useful for modeling the stego patches corresponding to more stego pixels.

The structure of the proposed MCNN is illustrated in Figure 1. It contains three parallel columns CNNs, all columns have the same network structures except for the sizes and numbers of filters. Note that the pooling operation is removed from all layers, and ReLU is adopted as an activation function in each convolution layer because of its good performance for CNNs [12]. To reduce the computational cost, less number of filters for CNNs with larger filters is used. We stack the output feature maps of each columns and adopt a 1×1 convolution layer to map the stacked output maps to the stego map before fully connected layer, which makes the input image be of arbitrary size so that the proposed method is adaptive to large variation in stego image size.

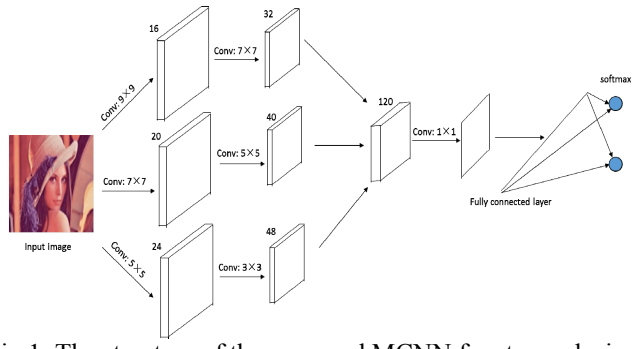

Fig.1: The structure of the proposed MCNN for steganalysis

The benefits of the designed MCNN structure include: 1) the proposed structure can deal with arbitrary size of input image without resizing, which is preferred in the fields of stegnalysis because image resizing will destroy the pixels relationship of the stego image. 2) the output of our MCNN is combined with learnable weights by 1×1 filters, which can capture the suitable features for steganalysis.

C. Optimization of MCNN training[12]

The softmax output can be optimized via batch-based stochastic gradient descent and back-propagation, typical for the training stage. However, as the number of training samples are very limited, it is not easy to learn all the parameters simultaneously. Motivated by the success of pre-training, CNN in each single column is pre-trained separately by directly mapping the outputs of the last convolutional layer to the stego map. Then three pre-trained CNNs are used to initialize CNNs in all columns and fine-tune all the parameters simultaneously.

IV. Experimental Results

A. Databases and parameters used

We evaluated the proposed MCNN-based steganalysis on two image cover databases, BOSS database [14] and RAISE database [15]. For the experiments, the BOSS database consists of 10000 grey-level images of size 512×512, and each image is divided into four parts to obtain 40000 images of size 256×256. The RAISE database includes 8156 high-resolution raw images, in which all photos are stored in uncompressed formats, in high quality (3008 × 2000, 4288 × 2848 and 4928 × 3264 pixels), and each image is also randomly divided to obtain 20000 images of size 512×512 and 20000 images of size 256×256. Note that we transformed high-resolution raw images in RAISE into grey-level images with the same parameters as in the BOSS.

In our evaluation, we embedded the messages using two steganographic tools of HUGO[16] and S-UNIWARD [17] with two embedding payload values: 0.4 and 0.1 bpp, and using the C++ implementations available from DDE Lab Binghamton web site. After embedding, the experimental database consisted 150000 images (50000 covers and 100000 stegos) from BOSS and 120000 images (40000 covers and 80000 stegos) from RAISE. We limited the experiments to these two payload in consideration of the high number of images and computations.

In our experiments, the training parameters were set as follows: the "mini-batch" size was set to 100, the "moment" was set to 0.01, the learning coefficient was set to 0.0001 for weights and set to 0.0002 for bias, the weight decay was set to 0.002 for convolutions layers and set to 0.001 for the fully connected network, learning method was set as "SGD", the "drop out" and "pooling" is not activated, the max training epochs were set as 200. With a learning database consisting of 200000 blended images of sizes 256×256 and 512×512. It took about five days to do computation in order to find the "best" network with the Nvidia Tesla K40 GPU card. Note that all experiments were done using Caffe framework.

B. Clairvoyant scenarios

In this scenario, we randomly select the cover and stego images from the same database, and the embedding algorithm

and the payload size are known by the steganalyst. That is to say, the steganalyst knows all the public parameters except the private parameters such as the embedding secret key. In order to compare with other CNN-based steganalyzer, we also additionally add the condition of stego images with the same key, which uses the same embedding key to apply steganography.

The experiments were all done on the BOSS database, which consists of 10000 8bits grey-level images with size of 512×512 and 10000 8bits grey-level images with size of 256×256. We embedded the messages using HUGO and S-UNIWARD with two embedding rate of 0.4 bpp and 0.1 bpp. So the experimental set of images was composed of 100000 images (20000 covers and 80000 stegos).

Three CNN-based steganalyzer methods were evaluated. The first steganalyzer was done using the CNN presented by Pibre et al. [10], which is denoted as Pibre-CNN. The second steganalyzer was done using the CNN presented by Couchot et al. [11], which is denoted as Couchot-CNN. The third steganalyzer was done using the MCNN we built (see Figure.1), which is denoted as MCNN.

For each payload size (0.4 bpp and 0.1 bpp) and mixed payload size, 10 tests were carried out where, for each test, the training was carried out on 40000 randomly selected images in the experimental image set. The tests were also carried out on 10000 randomly selected images in the remaining experimental image set (5000 covers and 5000 corresponding stegos).

For both Pibre-CNN, Couchot-CNN, and MCNN, detection accuracy was got by averaging over the 10 tests. The results are shown in Table 1. It is important to point out that the MCNN network usually converges before the end of the training iteration except the mixed steganographic algorithm and payload.

Table 1. Results under Clairvoyant scenarios

Stego algorithm	Payload	Iteration number	Detection accuracy		
			Pibre-CNN	Couchot-CNN	MCNN
HUGO	0.4	52	93.03%	93.76%	94.83%
HUGO	0.1	168	72.17%	75.59%	77.07%
S-UNIWARD	0.4	61	91.47%	93.15%	95.13%
S-UNIWARD	0.1	142	70.35%	72.94%	74.25%
Mixed	0.1&0.4	200	79.49%	80.14%	83.68%

Whatever the stegnanographic algorithm or payloads are chosen, the proposed MCNN has at least 1% improvement in detection accuracy compared with Pibre-CNN and Couchot-CNN. Note that there is at least 3.5% improvement in the situation of mixed steganographic algorithm and mixed payload, which is an impressive outperformance considering the improvement on the accuracy of steganalysis.

Our experiment shows that our proposed MCNN is trained well. We conjecture a few reasons:

i) There are three parallel CNN columns with different numbers of filters and different sizes of receptive fields. A large number of filters enriches the diversity of features extraction. The three different receptive fields have multi-scale analysis functions and can extract multi-scale features which improves the ability of characterizing local features.

ii) The pooling layers are eliminated because they are, by its very nature, a down-sampling process which may lead to information loss.

C. *Cover-source mismatch scenarios*

In this scenario, the trained networks in the previous section are chosen and applied to the testing image set, which consist of randomly selected 20000 images (10000 covers and 10000 stegos) in RAISE. The reason of selecting RAISE as cover-source mismatch experiments is that the RAISE has high-quality raw images with uncompressed formats.

The detection accuracy is reported in Table 2. We also evaluate the same three CNN-based steganalyzer methods as in the previous section. The results show that all three CNN-based steganalyzers with small payload training can detect high payload stego. In detail, the proposed MCNN has detection accuracy higher than 90% for the payload of 0.4bpp, while it falls to at least 65% for the payload of 0.1bpp and at least 81% for the unmatched training and testing payload. In comparison with the Pibre-CNN and Couchot-CNN, the proposed MCNN has about 2.5% improvement in detection accuracy.

Table 2. Results under Cover-source mismatch scenarios

Stego algorithm & payload		Iter number	Detection accuracy		
Training (BOSS)	Testing (RAISE)		Pibre-CNN	Couchot-CNN	MCNN
HUGO 0.4	HUGO 0.4	52	89.72%	88.03%	92.83%
HUGO 0.1	HUGO 0.1	168	65.46%	64.31%	68.87%
HUGO 0.1	HUGO 0.4	52	79.87%	81.34%	84.56%
S-UNI 0.4	S-UNI 0.4	61	87.02%	87.97%	90.45%
S-UNI 0.1	S-UNI0.1	142	61.45%	62.34%	65.05%
S-UNI 0.1	S-UNI 0.4	61	74.76%	78.21%	81.04%
Mixed	Mixed	200	70.49%	72.14%	74.73%

V. CONCLUSION

The increasing attention gained by deep learning has raised the interest in whether such a method is relevant for the design of steganalyzer [11]. In this paper, we put forward a MCNN-based steganalyzer. The designed MCNN structure consists of three parallel CNNs with different receptive fields, and each CNN consists only of two convolutional layers.

We evaluated the detection ability of the MCNN against two steganographiers of HUGO, and S-UNIWARD with payload of 0.1 bpp and 0.4 bpp in two standard databases of BOSS and Raise. The obtained results show the high performance of the proposed MCNN-based steganalyzer. More precisely, in comparison with the Ensemble Classifier with SRM features,

MCNN reduces the classification error by 10 percent or more. Also, compared to the previous CNN-based proposals for steganalysis, i.e., Pibre et al. and Couchot et al., MCNN improves the classification accuracy about 3 percent.

Since most of stegnographic softwares in Internet adopt the steganographic methods using the same secret embedding key, in which the stego software always uses the same pseudo-random number sequence to generate embedding positions, the proposed MCNN steganalysis is suitable for detecting internet stegnographic softwares.

In future work, we will concentrate on enlarging the set of steganography algorithms considered during both the training and the testing stages, and further experiments on different payload sizes. Moreover, our intension is to optimize some of the network parameters such as the shape of the filters, the columns of the CNNs, and to gain insight into the relationship between the CNNs and the chosen steganographiers.

ACKNOWLEDGMENT

This research was supported by natural science foundation of Guangdong Province (2017A030313374), scientific and technological projects of Guangzhou (201707010283), and scientific research project of Guangzhou municipal universities (2018A056).

REFERENCES

[1] V. Holub, J. Fridrich. Random projections of residuals for digital image steganalysis. IEEE Trans. information Forensics and Security, 8(12):1996-2006, 2013.

[2] Denemark, Tomáš, J. Fridrich, V. Holub. Further study on the security of S-UNIWARD. SPIE 9028, Electronic Imaging, Media Watermarking, Security, and Forensics, 2014.

[3] V. Holub, J. Fridrich. Low-complexity features for jpeg steganalysis using undecimated dct. IEEE Transactions on Information Forensics and Security, 10(2):219-228, Feb 2015.

[4] V. Holub, J Fridrich. Phase-aware projection model for steganalysis of JPEG images. SPIE 9409, Media Watermarking, Security, and Forensics 2015.

[5] Jan Kodovsky, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. IEEE Transactions on Information Forensics and Security, 7(2):432-444, 2012.

[6] Hans Georg Schaathun. Support Vector Machines, pages 179-196. John Wiley & Sons, Ltd, 2012.

[7] Ivans Lubenko and Andrew D. Ker. Steganalysis with mismatched covers: Do simple classifiers help? In Proceedings of the on Multimedia and Security, MM&Sec '12, pages 11-18, New York, NY, USA, 2012. ACM.

[8] Tan S Q, Li B. Stacked convolutional auto-encoders for steganalysis of digital images. In Asia-Pacific Signal and Information Processing Association, 2014 Annual Summit and Conference (APSIPA), pages 1-4, IEEE, 2014.

[9] Qian Y L, Dong J, Wang W, Tan T N. Deep learning for steganalysis via convolutional neural networks. In IS&T/SPIE Electronic Imaging, pages 94090J-94090J. International Society for Optics and Photonics, 2015.

[10] Pibre L, Jerome P, Ienco D, Chaumont M. Deep learning is a good steganalysis tool when embedding key is reused for different images, even if there is a cover source-mismatch. In EI: Electronic Imaging, 2016.

[11] Couchot J F, Couturier R, Guyeux C, Salomon M (2016) Steganalysis via a convolutional neural network using large convolution filters for embedding process with same stego key. arXiv:1605.07946v3

[12] Zhang Y Y, Zhou D S, et al. Single-image crowd counting via multi-column convolutional neural network, CVPR 2016: 589-597

[13] Zeiler M D, Ranzato M, Monga R, et al. On rectified linear units for speech processing. Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on. IEEE, 2013: 3517-3521.

[14] Pevný T, Filler T, Bas P. Using high-dimensional image models to perform highly undetectable Steganography. International Workshop on Information Hiding. Springer Berlin Heidelberg, 2010: 161-177.

[15] Dang-Nguyen D T, Pasquini C, Conotter V, et al. RAISE: a raw images dataset for digital image forensics. Proceedings of the 6th ACM Multimedia Systems Conference. ACM, 2015: 219-224.

[16] Pevný T, Filler T, Bas P. Using high-dimensional image models to perform highly undetectable steganography. International Workshop on Information Hiding. Springer Berlin Heidelberg, 2010: 161-177.

[17] V. Holub, Fridrich J, Denemark T. Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security,2014(1):1